16

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/622,413	07/18/2003	Erik Meijer	MSFT-1972/304029.1	4186

41505 7590 03/24/2006

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
ONE LIBERTY PLACE - 46TH FLOOR
PHILADELPHIA, PA 19103

EXAMINER	
CHEN, KOU-YI	
ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 03/24/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/622,413	**Applicant(s)** MEIJER ET AL.
	Examiner Kou-Yi K. Chen	**Art Unit** 2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)□ Responsive to communication(s) filed on *18 July 2003*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *18 July 2003* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *7/18/2003*.

4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed on July 18, 2003.

2. Claims 1-29 are pending in the application.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claim 1-3, 5-6, and 8-29 are rejected under 35 U.S.C. 102(e) as being anticipated by de Jong U.S. Publication No. 20040143827 (hereinafter, "de Jong").

As per claim 1
de Jong teaches:

1. A computer-readable medium having stored thereon a data structure ("data structures" in [0179] line 1) comprising:

a first field ("a value of a type identifier" in [0179] line 9) comprising data representing a linkage between accessibility and override-ability ("override method" in [0179] line 5) of a virtual method ("linking a program including virtual methods" in [0179] lines 1-2); and a second data field ("the address of the code" in [0179] lines 15-16) comprising data representing the accessibility ("corresponding to the virtual method instance" in [0179] line 16) of the virtual method;

wherein, if an attempt to override the virtual method is encountered, the first field is examined to determine the linkage ("the capability to link virtual methods" in [0178] line 2) and the second data field is optionally examined to verify accessibility ("the address of the appropriate virtual method code is obtained" in [0178] lines 11-12) before granting an override of the virtual method.

As per claim 2, the rejection of claim 1 is incorporated, and further, de Jong teaches: wherein the first field comprises at least one bit of metadata ("the type identifier associated with the instance is the value one" in [0180] lines 7-8).

As per claim 3, the rejection of claim 1 is incorporated, and further, de Jong teaches: wherein the second data field is examined ("determine which virtual method instance to call" in [0180] lines 10-11) if the first data field indicates that accessibility need be checked ("the type identifier is used as an index into the jump table" in [0180] lines 9-10) before granting an override of the virtual method.

As per claim 5

de Jong teaches:

A method of linking accessibility ("linking a program including virtual methods" in [0179] lines 1-2) and override-ability of virtual methods ("override method" in [0179] line 5) comprising:

setting at least one bit in metadata ("the check bit comprises the most significant bit" in [0210] line 2) indicative of allowing a virtual method to be overridden; and

checking the status of the at least one bit to determine ("a determination is made regarding whether the virtual method has been overridden with an instance of the virtual method" in [0182] lines 10-12) if accessibility of the virtual method need be verified before overriding the virtual method ("determine whether the method being called has been verified" in [0207] lines 10-11).

As per claim 6, the rejection of claim 5 is incorporated, and further, de Jong teaches: wherein a first state of the at least one bit is indicative of granting an unconditional override of the virtual method and a second state of the at least one bit is indicative of granting an override dependent on the accessibility of the virtual method ("a determination is made regarding whether the virtual method has been overridden with an instance of the virtual method. If the virtual method has been overridden, at 3725 a virtual method jump table for the virtual method having at least one virtual method instance is created" in [0182] lines 10-15).

As per claim 8

de Jong teaches:

A method of assessing permission to use a derived virtual method comprising:

receiving input code ("receiving a program" in [0021] lines 1-2) comprising at least one derived virtual method ("Linking a virtual method" in [0021] line 1) based on at least one virtual method of a programming language ("Java programming language" in [0011] line 2) utilized to generate the input code;

compiling the input code into at least one programming entity ("executable bytecodes" in [0121] line 5);

examining metadata ("the dispatch table entry corresponding to the called method is examined " in [0219] lines 23-24) to assess the permission to use the derived virtual method ("the check bit of the table entry has a value that indicates a checked status" in [0219] lines 25-26); and

performing one of granting use and denying use of the derived virtual method ("If the check bit value indicates an unchecked status, verification of the protection unit comprising the method is performed" in [0219] lines 29-31) dependent upon a link between accessibility and override-ability of the at least one virtual method ("a determination is made regarding whether a calling method and a called method are within the same protection unit" in [0219] lines 1-3);

wherein the link between accessibility and override-ability is established in the metadata ("the called method is called using the address in the table entry" in [0219] lines 33-34).

As per claim 9, the rejection of claim 8 is incorporated, and further, de Jong teaches: wherein the at least one programming entity comprises at least one of a module, an assembly ("executable bytecodes" in [0121] line 5, bytecode is an assembly) and an application.

As per claim 10, the rejection of claim 8 is incorporated, and further, de Jong teaches: wherein the at least one programming entity is an intermediate language representation ("executable bytecodes" in [0121] line 5, bytecode is an intermediate language representation between the Java source code and the executables).

As per claim 11, the rejection of claim 8 is incorporated, and further, de Jong teaches: wherein the link comprises at least one bit indicating that override-ability is dependent on accessibility ("a determination is made regarding whether the virtual method has been overridden with an instance of the virtual method. If the virtual method has been overridden, at 3725 a virtual method jump table for the virtual method having at least one virtual method instance is created" in [0182] lines 10-15).

As per claim 12
de Jong teaches:

A computer system comprising:

a processor ("central processor" in [0091] line 5) which acts upon an input program ("receiving a program" in [0021] lines 1-2) comprising at least one virtual method ("linking a program including virtual methods" in [0179] lines 1-2);

a mechanism ("This mechanism " in [0210] line 7) establishing a potential correspondence between accessibility and override-ability of a virtual method comprising at least one bit of metadata ("the type identifier associated with the instance is the value one" in [0180] lines 7-8);

wherein the processor utilizes the mechanism to detect the potential correspondence between accessibility and override-ability before allowing the at least one virtual method to be overridden ("a determination is made regarding whether the virtual method has been overridden with an instance of the virtual method" in [0182] lines 10-12).

As per claim 13 the rejection of claim 12 is incorporated, and further, de Jong teaches: wherein the mechanism ("This mechanism " in [0210] line 7) permits the at least one virtual method to be overridden if the at least one virtual method is not accessible ("obtain the method address" in [0210] line 8).

As per claim 14, the rejection of claim 12 is incorporated, and further, de Jong teaches: wherein the mechanism equates accessibility and override-ability ("the least-significant bits of the dispatch table entry are used to call the routine" in [0210] line 5-7).

As per claim 15, the rejection of claim 12 is incorporated, and further, de Jong teaches: wherein a derived method being generated from the at least one virtual method is subject to the mechanism establishing a potential correspondence between accessibility and override-ability of a virtual method ("Having one check bit per entry allows the tables to be contiguous; referencing an entry requires just the base address of the table and an offset. Additionally, each entry includes a check bit because the corresponding routine may be the first routine within a protection unit to be called" in [0209] lines 1-5).

As per claim 16

de Jong teaches;

A computer system performing the method of:

receiving program code ("receiving a program" in [0021] lines 1-2) comprising a derived virtual method ("Linking a virtual method" in [0021] line 1), the derived virtual method being derived from a virtual method of a program language ("Java programming language" in [0011] line 2) used to generate the program code;

compiling the program code to an intermediate language expression ("executable bytecodes" in [0121] line 5);

generating metadata descriptive of the compiled program code, the metadata indicating a linking of accessibility and override-ability of the virtual method ("linking a program including virtual methods" in [0179] lines 1-2);

accessing and examining the metadata ("the dispatch table entry corresponding to the called method is examined" in [0219] lines 23-24); and allowing the use of the derived virtual method based on the linking of accessibility and override-ability of the virtual method as indicated by the metadata ("Having one check bit per entry allows the tables to be contiguous; referencing an entry requires just the base address of the table and an offset" in [0209] lines 1-3).

As per claim 17, the rejection of claim 16 is incorporated, and further, de Jong teaches: wherein accessing and examining the metadata occurs in a common language infrastructure environment supporting at least one programming language ("In a Java environment" in [0108] line 7).

As per claim 18, the rejection of claim 16 is incorporated, and further, de Jong teaches: wherein the generating of metadata comprises generating at least one bit wherein the bit indicates a need to check accessibility before overriding a virtual method ("the type identifier associated with the instance is the value one" in [0180] lines 7-8).

As per claim 19, the rejection of claim 16 is incorporated, and further, de Jong teaches: wherein generating metadata comprises generating metadata while compiling the program code ("Method code data 1320 may comprise, by way of example, executable bytecodes. Field definition data 1325 may comprise, by way of example, a field count, and a field type identifier for each field included in the field count. Field initialization data 1330 may comprise, by way of example, data used to initialize constant data" In [0121] lines5-10).

As per claim 20, the rejection of claim 16 is incorporated, and further, de Jong teaches: wherein generating metadata comprises utilizing pre-existing values indicative of the access and override permissions concerning the virtual method ("a determination is made regarding whether the virtual method has been overridden with an instance of the virtual method" in [0182] lines 10-12) of the programming language.

As per claim 21
de Jong teaches:

A computer-readable medium having computer-executable instructions for performing a method comprising:

receiving program code ("receiving a program" in [0021] lines 1-2) comprising a derived virtual method, the derived virtual method ("Linking a virtual method" in [0021] line 1) being derived from a virtual method of a program language ("Java programming language" in [0011] line 2) used to generate the program code;

compiling the program code to an intermediate language expression ("executable bytecodes" in [0121] line 5);

generating metadata descriptive of the compiled program code, the metadata indicating a linking of accessibility and override-ability of the virtual method ("linking a program including virtual methods" in [0179] lines 1-2);

accessing and examining the metadata ("the dispatch table entry corresponding to the called method is examined" in [0219] lines 23-24); and

allowing the use of the derived virtual method based on the linking of accessibility and override-ability of the virtual method ("Having one check bit per entry allows the tables to be contiguous; referencing an entry requires just the base address of the table and an offset" in [0209] lines 1-3).

As per claim 22, the rejection of claim 21 is incorporated, and further, de Jong teaches: wherein the generating of metadata comprises generating at least one bit wherein the bit indicates the availability of overwriting the virtual method ("the type identifier associated with the instance is the value one" in [0180] lines 7-8).

As per claim 23, the rejection of claim 21 is incorporated, and further, de Jong teaches: wherein generating metadata comprises generating metadata while compiling the program code ("Method code data 1320 may comprise, by way of example, executable bytecodes. Field definition data 1325 may comprise, by way of example, a field count, and a field type identifier for each field included in the field count. Field initialization data 1330 may comprise, by way of example, data used to initialize constant data" In [0121] lines5-10).

As per claim 24 the rejection of claim 21 is incorporated, and further, de Jong teaches: wherein generating metadata comprises utilizing pre-existing values indicative of the access and override permissions concerning the virtual method ("a determination is made regarding whether the virtual method has been overridden with an instance of the virtual method" in [0182] lines 10-12) of the programming language.

As per claim 25
de Jong teaches:

A computer-readable medium having computer-executable instructions for performing a method of linking accessibility and override-ability of virtual methods comprising:
setting at least one bit in metadata indicative of allowing a virtual method to be overridden ("the type identifier associated with the instance is the value one" in [0180] lines 7-8); and
checking the status of the at least one bit to determine if accessibility of the virtual method need be verified before overriding the virtual method ("a determination is made regarding whether the virtual method has been overridden with an instance of the virtual method" in [0182] lines 10-12).

As per claim26, the rejection of claim 25 is incorporated, and further, de Jong teaches: wherein a first state of the at least one bit is indicative of granting an unconditional override of the virtual method and a second state of the at least one bit is indicative of granting an override dependent on the accessibility of the virtual method ("a determination is made regarding whether the virtual method has been overridden with an instance of the virtual method. If the virtual method has been overridden, at 3725 a virtual method jump table for the virtual method having at least one virtual method instance is created" in [0182] lines 10-15).

As per claim 27
de Jong teaches:

A computer-readable medium having computer-executable instructions for performing a method of assessing permission to use a derived virtual method comprising:

receiving input code ("receiving a program" in [0021] lines 1-2) comprising at least one derived virtual method ("Linking a virtual method" in [0021] line 1) wherein at least one derived virtual method is based on at least one virtual method of a programming language ("Java programming language" in [0011] line 2) utilized to generate the input code;

compiling the input code into at least one programming language ("executable bytecodes" in [0121] line 5);

examining metadata ("the dispatch table entry corresponding to the called method is examined " in [0219] lines 23-24) to assess the permission to use the derived virtual method ("the check bit of the table entry has a value that indicates a checked status" in [0219] lines 25-26); and

performing one of granting use and denying use of the derived virtual method ("If the check bit value indicates an unchecked status, verification of the protection unit comprising the method is performed" in [0219] lines 29-31) dependent upon a link between accessibility and override-ability of the at least one virtual method ("a

determination is made regarding whether a calling method and a called method are within the same protection unit" in [0219] lines 1-3);
wherein the link between accessibility and override-ability is established in the metadata ("the called method is called using the address in the table entry" in [0219] lines 33-34).

As per claim 28, the rejection of claim 27 is incorporated, and further, de Jong teaches: wherein the at least one programming language is an intermediate language representation ("executable bytecodes" in [0121] line 5, bytecode is an intermediate language representation between the Java source code and the executables).

As per claim 29, the rejection of claim 27 is incorporated, and further, de Jong teaches: wherein the link comprises at least one bit indicating that override-ability is dependent on accessibility based on the status of the bit ("a determination is made regarding whether the virtual method has been overridden with an instance of the virtual method. If the virtual method has been overridden, at 3725 a virtual method jump table for the virtual method having at least one virtual method instance is created" in [0182] lines 10-15).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and

> the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 4 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over de Jong in view of US Patent 6,487,713 to Cohen et al. (hereinafter, "Cohen").

As per claim 4, the rejection of claim 1 is incorporated, and further, de Jong teaches a bitmapped class-level 4-bit type may be used to represent the 4 most-used types in a class (see de Jong [0130] lines 11-12), but does not explicitly teach the field comprises at least one of a private bit, a public bit, an assembly bit, a family bit and a virtual bit. However, Cohen discloses in an analogous system setting the TYPE_PUBLIC bit to indicate that it is a public procedure (see Cohen col. 56 lines 47-48). It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to modify the teachings of de Jong with setting TYPE_PUBLIC bit as taught by Cohen. The modification would have been obvious because one of the ordinary skill in the art would have been motivated to indicate that the public procedure or label has been declared (see Cohen col. 56 lines 31-32) in de Jong's disclosed system, using Cohen's suggestion in col. 56 lines 47-48.

As per claim 7, the rejection of claim 5 is incorporated, and further, de Jong teaches: wherein accessibility of the virtual method comprises flag bits in metadata comprising a private bit, a public bit, an assembly bit, a family bit and a virtual bit (see claim 4 rejection).

Conclusion

7. The prior art made of record, and not relied upon, is considered pertinent to applicant's disclosure.

Contact Information

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Kou-Yi K. Chen** whose telephone number is **571-272-8592**. The examiner can normally be reached **from 8:30 am to 5:00 pm on M-F**.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on 571-272-3719. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kou-Yi K. Chen
Assistant Examiner
Art Unit 2193

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100